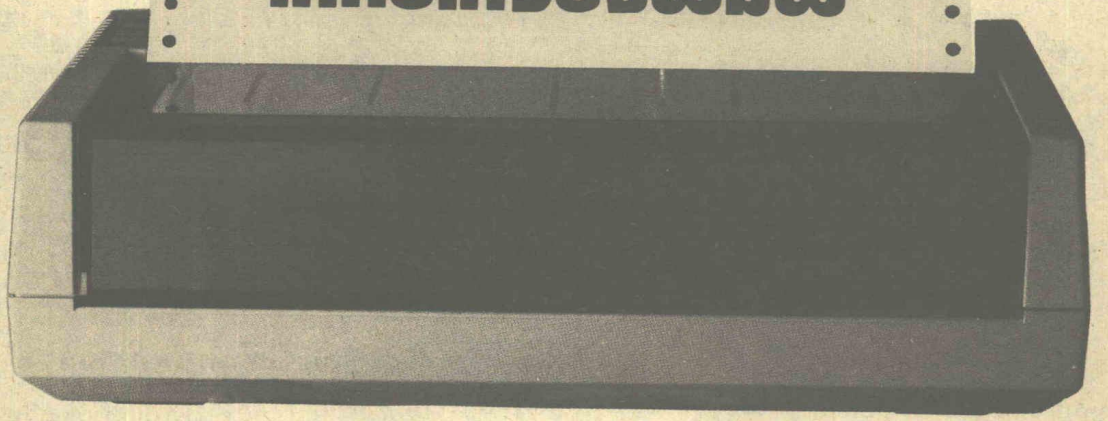
# สร้างตัวหนังสือไทยให้กับเครื่องพิมพ์



โปรแกรมนี้ใช้กับเครื่องไอบีเอ็ม

ซอฟท์แวร์นี้สงวนลิขสิทธิ์
ตาม พ.ร.บ.ลิขสิทธิ์ พ.ศ. 2521
ห้ามลอกเลียนไม่ว่าส่วนหนึ่งส่วนใด
ของซอฟท์แวร์นี้
**เพื่อใช้ประโยชน์ในทางการค้า**
นอกจากจะได้รับอนุญาต

**จเร เลิศสุดวิชัย***

ในการออกแบบตัวหนังสือภาษาไทยให้กับเครื่องพิมพ์ในปัจจุบันทำได้ 2 แบบ คือ วิธีแรกเขียนรูปแบบแล้วเปลี่ยนเป็นรหัสตัวเลขบรรจุลงใน ROM เพื่อที่จะแทน ROM ตัวเดิม แต่วิธีนี้ผู้ที่มีความรู้ทางฮาร์ดแวร์เท่านั้นที่จะทำได้เพราะต้องทำทั้งการเบิร์นอีพรอมและการถอดเปลี่ยน ที่สำคัญคือ กว่าจะกำหนดรูปร่างหรือหาตำแหน่งเดิมของอีพรอมได้ต้องเสียเวลาพอควร วิธีที่สองนี้ใช้ได้กับเครื่องพิมพ์ที่สามารถดาวน์โหลดรูปลักษณะตัวหนังสือได้ซึ่งในปัจจุบันเครื่องพิมพ์ดอต-เมทริกซ์ส่วนใหญ่จะรับการดาวน์โหลดได้อยู่แล้ว ดังนั้นการใช้วิธีนี้จึงไม่ยุ่งยากในเรื่องการถอดรื้อเครื่อง แต่มีข้อเสียคือเมื่อต้องการใช้ตัวอักษรไทยจะต้องโหลด font ตอนเปิดเครื่องพิมพ์ทุกครั้งจะรู้สึกว่าไม่สนุกเท่าไร โดยเฉพาะถ้าที่บ้านใครไฟกะพริบบ่อย ๆ ละก็ ตัวหนังสือหายไปได้ แต่อย่างไรก็ตามวิธีนี้ก็มีความคล่องตัวสูง เพราะผู้ใช้สามารถเปลงตัวหนังสือเองได้

## ที่มาของโปรแกรมสร้างตัวอักษรไทย

ในการใช้ภาษาไทยนั้น น่าจะมีคนสร้าง Thai Toolbox สำหรับเทอร์โบปาสคาล เพื่อใช้ประโยชน์ทางด้านภาษาไทย โดยโปรแกรมในทูลบล็อกนี้จะประกอบด้วยโปรแกรมจัดการภาษาไทยหลายอย่าง ทั้งทางการแสดงออกจอเครื่องพิมพ์ และการประมวลผลข้อมูลภาษาไทยที่จะให้ผู้ใช้เลือกใช้ได้ ผู้เขียนจึงคิดว่าการสร้างโปรแกรมออกแบบตัวอักษรที่ใช้ง่ายๆ และก็ให้รายละเอียดในการสร้างด้วย เพื่อว่าจะได้ดัดแปลงแก้ไขโปรแกรมเอาเองได้ ในที่นี้จึงขอเลือกใช้ปาสคาลและก็เป็นผลในสิ่งที่เห็นนี้ โดยเลือกเครื่องพิมพ์กลุ่ม FX ก่อน แต่ถ้าเป็นเครื่องพิมพ์อื่นก็ดัดแปลงในโปรแกรมได้ไม่ยาก

## ขีดความสามารถของโปรแกรม

โปรแกรมออกแบบตัวอักษรนี้จะต้องใช้งานได้ง่าย โดยออกแบบตัวอักษรได้ทุกตัวตั้งแต่รหัส 160 ถึง 255 โดยใช้คีย์บอร์ดควบคุมเหมือนการเล่นเกมเป็นแบบฟูลสกรีน ฟังก์ชันของโปรแกรมควรทำได้ดังนี้

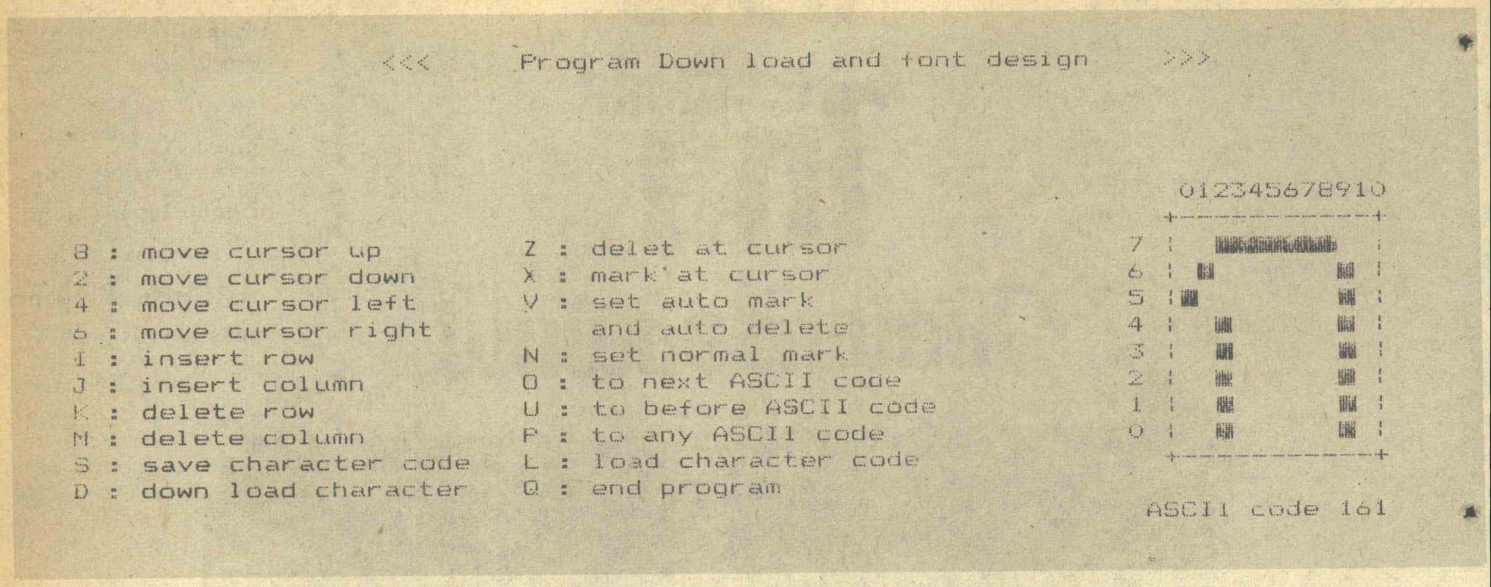
1. เลื่อนไปได้แบบฟูลสกรีน
2. มาร์คจุด ลบจุด ณ ตำแหน่งใดได้ง่าย
3. แทรกแถวหรือคอลัมน์ได้
4. ลบแถวหรือคอลัมน์ได้
5. อ่านไฟล์ข้อมูลตัวหนังสือมาแก้ได้ หรือ save ข้อมูลที่แก้ไขแล้วได้
6. นำข้อมูลตัวหนังสือโหลดเข้าเครื่องพิมพ์ได้

และด้วยขีดความสามารถนี้เอง จึงสร้างเป็นเมนูที่ปรากฏบนจอภาพได้ดังรูปที่ 1

## ลักษณะของ Font

ปกติหัวของเครื่องพิมพ์จะจัดพิมพ์ทางด้านแนวดิ่งโดยพิมพ์เรียงทีละคอลัมน์ ตัวหนังสือแต่ละตัวจะต้องประกอบด้วยข้อมูล 12 ไบต์ โดยเป็นตัวอักษรจริงๆ เพียง 9 ไบต์ ที่เหลือไบต์แรกเป็นแอตตริบิ้ว และสองไบต์สุดท้ายจะเป็น 00 โดยลักษณะของข้อมูลที่ใช้จะเป็นดังนี้

* ภาควิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเกษตรศาสตร์




**รูปที่ 1 แสดงเมนูโปรแกรมสร้างอักษรไทย**

## แอตตริบิ้ว (attribute)

จะเป็นข้อมูล 1 ไบต์ ซึ่งประกอบด้วย

● บิทที่ 1–4 จะกำหนดคอลัมน์ท้ายสุดของตัวอักษรมีค่ากำกับดังนี้

| คอลัมน์ | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| ตัวเลข | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |

● บิทที่ 5–7 จะกำหนดคอลัมน์แรกของตัวอักษรมีค่ากำกับดังนี้

| คอลัมน์ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| ตัวเลข | 0 | 16 | 32 | 48 | 64 | 80 | 96 | 112 |

● บิทสุดท้าย จะกำหนดการใช้เข็มที่ 1 ของหัวเครื่องพิมพ์

ถ้าใช้เข็มที่ 1 บิท นี้มีค่าเป็น 0

ถ้าไม่ใช้เข็มที่ 1 บิท นี้มีค่าเป็น 1

เช่น ถ้าเราไม่ใช้เข็มที่ 1 คอลัมน์แรกและสุดท้ายของตัวอักษรคือ 0 และ 9 ดังนั้น

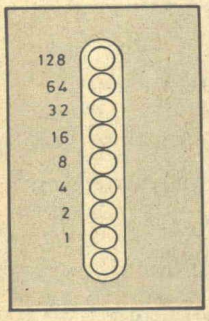
แอตตริบิ้ว = 128 + 0 + 11

= 139

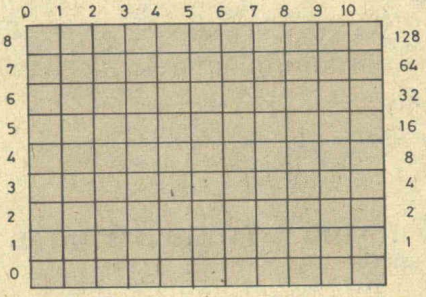
ปกติแล้วหัวพิมพ์ของเครื่องพิมพ์จะมี 9 เข็ม แต่ละเข็มมีค่าประจำของมันเอง ดังรูปที่ 2 ในที่นี้เราจะใช้เพียง 8 เข็ม ก็จะได้ข้อมูล 1 ไบท์ ต่อหนึ่งคอลัมน์

เช่นถ้าเราจะให้เข็มที่ 2, 4, 7 กระแทกให้เกิดจุด ข้อมูลไบต์นั้นจะมีค่าเท่ากับ 1+4+32 = 37

รายละเอียดของ 1 ตัวอักษรจะใช้ข้อมูล 10 ไบต์ เพื่อให้เกิดเข็มกระแทก ดังนั้นเมื่อรวมแอตตริบิ้ว ก็จะเป็น 11 ไบต์ต่อหนึ่งตัวอักษร ดังรูปที่ 3




**รูปที่ 2 แสดงเข็มของหัวเครื่องพิมพ์ เข็มที่ 1(ล่างสุด) จะไม่ใช้ ทุกเข็มจะมีค่าประจำเข็ม**




**รูปที่ 3 แสดงขอบเขตของตัวอักษรที่จะสร้างขึ้น**

## ขั้นตอนการสร้างตัวหนังสือไทย

1. เมื่อเราเปิดสวิทช์เครื่องพิมพ์ หน่วยความจำ RAM ในเครื่องพิมพ์จะมีเนื้อที่เหลือพอสำหรับที่เราจะสร้างตัวอักษรได้ เพื่อที่เราจะได้ใช้ตัวอักษรใน ROM พร้อมๆ กับตัวอักษรที่เราสร้างขึ้นมา เราจะต้องก๊อบปี้หรือถ่ายทอดเอาอักษรบน ROM มาไว้ใน RAM ก่อน คำสั่งที่จะต้องให้เครื่องพิมพ์เพื่อก๊อบปี้ ROM มาไว้ใน RAM ก็คือ

**writeln (1st, chr(27), ':', chr(o), chr(o), chr(o));**

โดยที่ Chr(27), ':', chr(0), chr(0), chr(0) ก็คือรหัสควบคุมที่สั่งให้เครื่องพิมพ์ทำการก๊อบปี้ตัวอักษรใน ROM มาไว้ ที่ RAM

2. หลังจากนั้นเราจะใช้หน่วยความจำ RAM แทน ROM โดยส่งรหัสควบคุมดังนี้

**writeln (1st, chr(27), '%', chr(n), chr(o));**

โดยที่ n = 0 ใช้หน่วยความจำ ROM

n = 1 ใช้หน่วยความจำ RAM

3. ทำการกำหนดตำแหน่งหรือ ASCII CODE ที่เราจะทำการสร้างตัวอักษรขึ้น

**writeln (1st, chr(27), chr(o), chr(C1), chr(C2));**

C1 และ C2 กำหนดได้ดังนี้

C1 เป็นตำแหน่งเริ่มต้นของตัวอักษรที่เราจะสร้าง

C2 เป็นตำแหน่งสุดท้าย ของตัวอักษรที่เราสร้าง

เช่น ภาษาไทยที่เราจะสร้างเริ่มที่ 160 ถึง 255

ดังนั้น C1 = 160

C2 = 255

4. ส่งข้อมูลของแต่ละตัวอักษรตามไป ซึ่งแต่ละตัวจะประกอบด้วยแอตตริบิ้วและรายละเอียดของตัวอักษร

## ตัวอย่างโปรแกรมสร้างอักษร ก.ไก่

เปิดสวิทช์เครื่องพิมพ์แล้วรันโปรแกรมตัวอย่างในรูปที่ 4 เราก็สามารถให้เครื่องพิมพ์ ก.ไก่ขึ้นที่ ASCII CODE 161 ตามรูปร่างที่เราสร้างขึ้นมาได้ดังรูปที่ 5 โดยที่ยังใช้ตัวอักษรภาษาอังกฤษที่มีอยู่แล้วได้เหมือนเดิมทุกประการ

แต่ถ้าเราจะสร้างอักษรไทยขึ้นมาสักชุดหนึ่ง โดยเรียงตั้งแต่ ASCII CODE 160-255 เราคงต้องพิมพ์ข้อมูลกันเมื่อยมือแน่ แถมการแก้ไขรูปร่างก็ยุ่งยากด้วย โปรแกรมที่ลิสต์มานี้จะช่วยในการสร้างตัวอักษรไทยได้สะดวกขึ้น โดยที่เราเพียงแต่เลื่อนเคอร์เซอร์ไปตามตำแหน่งที่ต้องการในกรอบสี่เหลี่ยมกว้าง 8 ช่องยาว 10 ช่อง แล้วทำการมาร์กจุดให้เป็นรูปตัวหนังสือที่ต้องการ ดูในเมนู

คีย์ที่ใช้ในการเลื่อนเคอร์เซอร์ใช้กลุ่มตัวเลขทางขวามือโดยกด number lock ก่อน

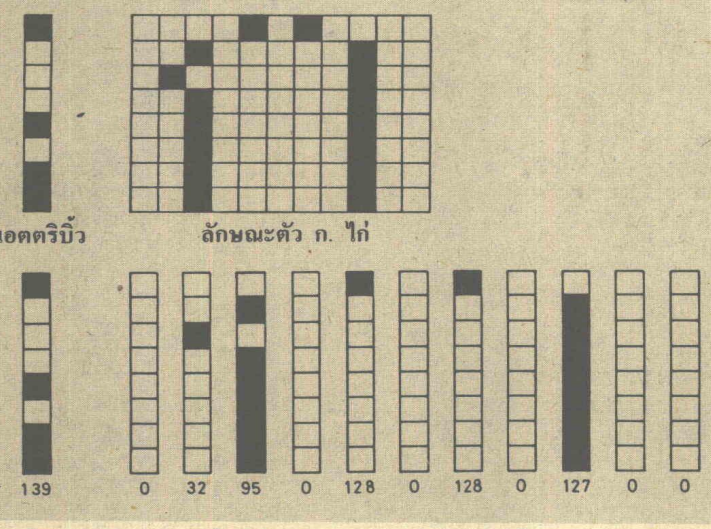
- "8" เลื่อนเคอร์เซอร์ขึ้น
- "2" เลื่อนเคอร์เซอร์ลง
- "4" เลื่อนเคอร์เซอร์ไปทางซ้าย
- "6" เลื่อนเคอร์เซอร์ไปทางขวา
- "I" แทรก คอลัมน์ตรงตำแหน่งเคอร์เซอร์
- "J" แทรก แถว
- "K" ลบ คอลัมน์
- "M" ลบ แถว
- "S" ทำการเก็บข้อมูลตัวอักษรที่เราออกแบบ ใส่ชื่อไฟล์ที่ต้องการเก็บโปรแกรม จะใส่นามสกุลให้เป็น .FXD
- "D" ใช้ดาวน์โหลดข้อมูลตัวอักษรลงเครื่องพิมพ์
- "Z" ลบจุดตรงตำแหน่งเคอร์เซอร์
- "X" มาร์กจุดตรงตำแหน่งเคอร์เซอร์
- "V" กดครั้งแรกจะมาร์กจุดให้อัตโนมัติขณะเลื่อนเคอร์เซอร์ไป กดอีกครั้ง จะลบจุดให้อัตโนมัติขณะเลื่อนเคอร์เซอร์ไป
- "N" ลบล้างการมาร์ก และลบจุดอัตโนมัติ
- "0" เพิ่มตำแหน่ง ASCII CODE 1 ตำแหน่ง
- "U" ลดตำแหน่ง ASCII CODE 1 ตำแหน่ง
- "P" ไปที่ตำแหน่ง ASCII CODE ใดๆ (160-255)
- "L" โหลดข้อมูลตัวอักษรมาทำการแก้ไขเปลี่ยนแปลง

## โปรแกรมการสร้างตัวอักษรไทยบนเครื่องพิมพ์

โปรแกรมที่ให้มานี้ใช้กับเครื่องพิมพ์รุ่น FX-80 ถ้าเป็นรุ่นอื่น ต้องดูในคู่มือเครื่องพิมพ์แล้วแก้ไขรหัสควบคุมในโปรแกรม ก็สามารถใช้โปรแกรมนี้เครื่องพิมพ์นั้นได้

```
Program creat_font;
begin
  { copy ROM to RAM }
  writeln(lst,chr(27),':',chr(0),chr(0),chr(0));
  { use RAM }
  writeln(lst,chr(27),'%',chr(1),chr(0));
  { at ASCII CODE 161 }
  writeln(lst,chr(27),chr(0),chr(161),chr(161));
  { Attibute and DATA }
  writeln(lst,chr(139),chr(0),chr(32),chr(95),
          chr(0),chr(128),chr(0),chr(128),chr(0),
          chr(127),chr(0),chr(0));
  writeln(lst,chr(161))
end.
```

รูปที่ 4 โปรแกรมสร้างอักษร ก ไก่ที่ ASCII CODE 161




รูปที่ 5 แสดงรายละเอียดของอักษรตัว ก.ไก่ และการให้ค่าของข้อมูลแต่ละคอลัมน์จะหมายถึงการทำให้เข็มแต่ละเข็มทำงาน

# โปรแกรมสร้างอักษรไทยบนเครื่องพิมพ์ FX-80

```
A>type fx80.pas
Program Dodw_load_and_font_design;
type    file_name          =       string[14];
        f                  =       file;
var      flag_mark,
         edit_flag,
         flag,exit         :      boolean;

        a,x,y              :       integer;
        font               :       array[160..255,1..11] of byte;
        chara,c            :       char;
        datafile           :       f;


function test_file(var tfile :f) : boolean;
begin
    {$i-} reset(tfile) ; {$i+}
    test_file :=(IOresult =0)
end;


procedure write_filename( filename  : file_name ; x,y : integer);
begin
    gotoxy(x,y);
    write(filename);
end;


procedure check_filename(var pfile : file_name);
begin
   if pos('.',pfile)=0 then;
      pfile :=concat(pfile,'.FXD');
end;



procedure directory;
type

   char12arr              = array[1..12] of char;
   string20               = string[20];
   regrec                 =
      record
         AX,BX,CX,DX,BP,SI,DI,DS,ES,Flags    : integer;
      end;

var
  regs                   : regrec;
  DTA                    : array[1..43] of byte;
  Mask                   : char12arr;
  Namr                   : string20;
  Error,I,j              : integer;

begin
    j :=1;
    Fillchar(DTA,SizeOf(DTA),0);
    Fillchar(Mask,Sizeof(Mask),0);
    Fillchar(Namr,Sizeof(Namr),0);
    writeln('Directory data file FX80 ');
    writeln;
    regs.AX :=$1A00;
    regs.DS :=Seg(DTA);
    regs.DX :=Ofs(DTA);
    MSDos(regs);
    Mask :='????????.FXD';
```

```
      regs.AX :=$4E00;
      regs.DS :=Seg(Mask);
      regs.DX :=Ofs(Mask);
     regs.CX :=22;
     MSDos(Regs);
     Error :=regs.AX and $FF;
     I :=1;
     if (Error=0) then
         repeat
         Namr[I] :=chr(Mem[Seg(DTA) :Ofs(DTA)+29+I]);
         I :=I+1;
      until not(Namr[I-1] in [' '..'~']) or (I>20);

      Namr[0] :=chr(I-1);

      while (Error =0) do begin
          Error :=0;
          regs.AX :=$4F00;
          regs.CX :=22;
          MSDos(regs);
          Error :=regs.AX and $FF;
          I :=1;
          repeat
             Namr[I] :=chr(Mem[Seg(DTA):Ofs(DTA)+29+I]);
             I :=I+1;
          until not(Namr[I-1] in [' '..'~']) or (I>20);
          Namr[0] :=chr(I-1);
          j :=1;
          if (Error =0) then begin
              write(Namr:12,'    ');
              if j=5 then begin
                 writeln;
                 j :=0
              end;
              j :=j+1;
          end
       end
end;


procedure plot_h;
var     i         :       integer;
begin
     for i :=1 to 11 do
         write('-');
end;

procedure plot_v(x,y  : integer);
var     i         :       integer;
begin
     for i :=1 to 8 do  begin
         gotoxy(x,y);
         write('¦');
         y :=y+1;
     end;
end;


procedure plot_corner;
begin
   gotoxy(65,9) ; write('+');
   gotoxy(77,9) ; write('+');
   gotoxy(65,18) ; write('+');
   gotoxy(77,18) ; write('+');
end;
```

```
procedure plot_border;
begin
begin
   plot_corner;
   gotoxy(66,9);
   plot_h;
   gotoxy(66,18);
   plot_h;
   plot_v(65,10);
   plot_v(77,10);
end;

procedure plot_pattern;
var j,i :integer ;
begin
    plot_border;
    gotoxy(66,8);
    for i :=0 to 10 do write(i);
    gotoxy(63,10);
    for i :=7 downto 0 do
        begin
            write(i);
            gotoxy(63,18-i)
        end;
end;


procedure menu;
begin
    gotoxy(15,3);
    write('     <<<     Program Down load and font design    >>>');
    gotoxy(1,10);
    writeln(' 8 : move cursor up          Z : delet at cursor');
    writeln(' 2 : move cursor down        X : mark at cursor');
    writeln(' 4 : move cursor left        V : set auto mark');
    writeln(' 6 : move cursor right           and auto delete');
    writeln(' I : insert row              N : set normal mark');
    writeln(' J : insert column           O : to next ASCII code');
    writeln(' K : delete row              U : to before ASCII code');
    writeln(' M : delete column           P : to any ASCII code');
    writeln(' S : save character code     L : load character code ');
    writeln(' D : down load character     Q : end program');
end;


procedure cursor(x,y : integer ; blink  : boolean);

var
   segment,offset    : integer;
begin
   segment := $b000;
   offset  := (160*(y-1)) + (x-1)*2;
   if blink then
      memw[segment:offset] := $87DB
   else
      begin
        gotoxy(x,y);
        write(chr(219))
      end;
   gotoxy(x,y);
end;


function testbit(n :byte; bit :integer) :boolean;
```

```
var i,j :byte ;
begin
     j :=$1;
     i :=j shl (bit-1);
     if (i and n) =0 then
          testbit :=false
     else
          testbit :=true
end;

procedure cursor_blink(blink : boolean);
begin
 if testbit(font[a,x-65],18-y) then
    cursor(x,y,blink)
end;

procedure inputkey;
begin
  read(kbd,chara);
end;

procedure on;
var t:byte;
begin
  write(chr(219));
  gotoxy(x,y);
  t:=$1;
  t:=t shl(17-y);
  font[a,x-65] :=font [a,x-65] or t;
end;

procedure off;
var t,t1,t2 : byte ;
begin
   write(' ');
   gotoxy(x,y);
   t :=$ff;
   t1 :=t shr (y-9);
   t :=$ff;
   t2 :=t shl (18-y);
   t :=t1 or t2;
   font[a,x-65] :=font[a,x-65] and t;
end;

procedure check;
begin
    if flag then begin
       if flag_mark then
          on
       else
          off;
    end;
end;

procedure move_up;
begin
   cursor_blink(false);
   check;
   y:=y-1;
   if y <= 9 then y:=17;
end;
```

```
procedure move_down;
begin
   cursor_blink(false);
   check;
   y:=y+1;
   if y>=18 then y :=10;
end;

procedure move_left;
begin
   cursor_blink(false);
   check;
   x:=x-1;
   if x<=65 then x :=75;
end;

procedure move_right;
begin
   cursor_blink(false);
   check;
   x :=x+1;
   if x>=77 then x :=66;
end;

procedure  plot( n :byte ; x: integer);
var i : integer;
begin
    for i :=1 to 8 do
        begin
            gotoxy(x,18-i);
            if testbit(n,i) then
               write(chr(219))
            else
               write(' ');
        end;
end;

procedure read_plot;
var i : integer;
begin
    for i :=1 to 11 do
        plot(font[a,i],65+i);
    gotoxy(64,20);
    write('ASCII code ',a:3);
    x :=66 ; y :=10 ;
end;

procedure insert_colum;
var t :byte;
    i :integer;
begin
     for i :=11 downto (x-65) do
         begin
             t :=font[a,i-1];
             font[a,i] :=t
         end;
     font[a,x-65] :=$0;
     read_plot
end;

procedure insert(var n :byte );
```

```
var t1,t2 :byte;
begin
     t1 :=$ff shr (y-10);
     t2 :=$ff shl (18-y);
     t1 :=t1 and n;
     t1 :=t1 shr 1;
     t2 :=t2 and n;
     n :=t1 or t2;
end;


procedure insert_row;
var i :integer;
begin
     for i :=1 to 11 do
        insert(font[a,i]);
     read_plot
end;


procedure delet_colum;
var i :integer;
     t :byte;
begin
     for i :=(x-65) to 11 do
         begin
             t :=font[a,i+1];
             font[a,i] :=t
         end;
     font[a,11] :=$0;
     read_plot
end;


procedure delet(var n:byte );
var t1,t2 :byte;
begin
     t1 :=$ff shr (y-9);
     t2 :=$ff shl (18-y);
     t1 :=t1 and n;
     t2 :=t2 and n;
     t1 :=t1 shl 1;
     n :=t1 or t2;
end;


procedure delet_row;
var i :integer;
begin
     for i :=1 to 11 do
         delet(font[a,i]);
     read_plot
end;


procedure move_to;
begin
     a :=a+1;
     if a >= 256 then a:=160;
     read_plot;
     x:=66;y :=10;
end;


procedure back_to;
```

```
begin
     a :=a-1;
     if a <=159 then a:=255;

    read_plot ;
    x :=66; y := 10;
end;


procedure ask;
begin
   gotoxy(20,22);
   write('Enter ASCII code : ');
   readln(a);
   if a <=159 then
      a :=255
   else
         if a >=256 then
            a :=160;
   gotoxy(20,22);
   write('                                ');
   read_plot;
   x :=66; y :=10;
end;


procedure setflag_mark;
begin
    flag_mark :=not flag_mark;
    flag :=true;
    gotoxy(66,5);
    if flag_mark then
          write('Auto Mark ')
    else
          write('Auto Delet')
end;


procedure setflag;
begin
    flag :=false;
    flag_mark :=false;
    gotoxy(66,5);
    write('            ')
end;


procedure check_flag;
begin
    gotoxy(66,5);
    if flag then
       begin
         gotoxy(66,5);
         if flag_mark then
            write('Auto Mark ')
         else
            write('Auto Delet');
       end;
end;


procedure make_array;
var   i,j   :  integer;
begin
    for i := 160  to 255 do
```

```
        for j :=1 to 11 do
            font[i,j] :=$0;
end;

  procedure plot_screen;
  begin
      clrscr;
      plot_pattern;
      check_flag;
      menu;
      read_plot;
      gotoxy(x,y);
  end;

  procedure save_newfile;
  begin
      rewrite(datafile);
      blockwrite(datafile,font,9);
      close(datafile);
  end;

  procedure save_oldfile;
  begin
      reset(datafile);
      blockwrite(datafile,font,9);
      close(datafile);
  end;

  procedure savel;
  var  sfile    :  file_name;
       c        :  char;
  begin
     clrscr;
     directory;
     gotoxy(10,12);
     write('Enter file name to save data : ');
     readln(sfile);
     check_filename(sfile);
     write_filename(sfile,41,12);
     assign(datafile,sfile);
     if test_file(datafile) then begin
        writeln;writeln;
        writeln('                        1. Save overwrite');
        writeln; writeln('                              2. Cancel');
        writeln; write('                 Select number : ');
        read(kbd,c);
        gotoxy(wherex-1,wherey);
        write(c);
        if c='1' then
           save_oldfile
     end
        else
           save_newfile;
  end;

  procedure wait;
  var    c    :  char;
  begin
      writeln; writeln;
      write('                          Save complete , press any key');
      read(kbd,c);
  end;

  procedure save;
  begin
```

```
        savel;
        wait;
        plot_screen;
    end;

procedure load_file;
begin
    reset(datafile);
    blockread(datafile,font,9)
end;

procedure load;
var    flag     :   boolean;
       filename : file_name;
begin
    flag :=false;
    clrscr;
    directory;
    repeat
        gotoxy(15,12);
        write('Enter file name to load : ');
        read(filename);
        check_filename(filename);
        write_filename(filename,41,12);
        assign(datafile,filename);
        if test_file(datafile) then  begin
               load_file;
               flag :=true;
           end
        else
           write_filename('                ',41,12);
     until flag;
     a :=160;
     plot_screen;
end;

procedure quit;
var c   : char;
begin
   gotoxy(15,21);
   write('Save data (Y/N) ? ');
   read(kbd,c);
   gotoxy(wherex-1,wherey);
   write(c);
   if upcase(c)='Y' then
       savel;
   edit_flag :=true;
   clrscr;
end;


procedure down_load_font;   { This procedure download printer }

procedure copy_rom; {  Copy character from rom to ram }
begin
     write(lst,#27,':',#0,#0,#0)
end;

procedure download_cg;   { Design character code from 160 to 255 }
var i,j :integer;
begin
    write(lst,#27,'%',#1,#0);
    write(lst,#27,'&',#0);
    write(lst,#160,#255);
    for i :=160 to 255 do
```

```
          begin
               write(lst,#139);
               for j :=1 to 11 do
                   write(lst,chr(font[i,j]))
          end;
end;

begin { main down_load_font }
     copy_rom;
     download_cg
end;


procedure test_print;    {  Test printer to print character }
var i,j:integer;
    ans :char;
begin
    ans :=' ';
    repeat
          gotoxy(10,20);
          writeln('Download  fontdesign complete');
          writeln;write('      Do you want to test printer ? (Y/N) ');
          read(kbd,ans);
          gotoxy(wherex-1,wherey);
          write(ans);
    until (upcase(ans) = 'Y') or (upcase(ans) ='N');
    if upcase(ans)='Y' then
          begin
              writeln(lst,#199,#196,#221,#199,#178,#218,#163,#193,#221,#184);
              for j :=1 to 3 do
                  begin
                     for i :=160 to 255 do
                        write(lst,chr(i))
                  end;
               writeln(lst,'  ');
          end;
end;


procedure download;  { put  charecter  to printer }
var i,j      :  integer;
    sfile    :  file_name;
    b        :  boolean;
    c        :  char;
begin
    b :=false;
    while not b do
    begin
      clrscr;
      directory;
      gotoxy(10,10);
      writeln(' This procedure use to download fontdesign to printer');
      writeln;
      write('                        Enter file name : ');
      read(sfile);
      check_filename(sfile);
      write_filename(sfile,41,12);
      assign(datafile,sfile);
      if test_file(datafile) then
        b :=true
      else
      begin
        writeln;writeln;writeln('       There has not file ','  :',sfile);
        writeln;writeln('       Enter file name agian');
        delay(1000);
```

```
          b :=false;
        end;
      end;
        writeln;writeln;
        writeln('        WARNING ! power on your printer');
        writeln;write('          (Press return when you ready) :');
        repeat
          read(kbd,c);
        until c =#13;
        reset(datafile);
        blockread(datafile,font,9);
        down_load_font;
        test_print;
        plot_screen;
end;


procedure testkey;
begin
    case upcase(chara) of
         '4' : move_left;
         '8' : move_up;
         '6' : move_right;
         '2' : move_down;
         'X' : on;
         'Z' : off;
         'J' : insert_colum;
         'I' : insert_row;
         'M' : delet_colum;
         'K' : delet_row;
         'U' : back_to;
         'O' : move_to;
         'V' : setflag_mark;
         'N' : setflag;
         'P' : ask;
         'L' : load;
         'S' : save;
         'D' : download;
         'Q' : quit;
     end;
end;

procedure initial_set;
begin
   clrscr;
   a :=160;
   flag_mark :=false;
   flag :=false;
   edit_flag :=false;
   x :=66 ; y :=10;
end;


begin  { main of procedure edit and design font }
     initial_set;
     make_array;
     plot_screen;
     repeat
          inputkey;
          testkey;
          gotoxy(x,y);
          cursor_blink(true);
     until edit_flag;
end.
```